ASM09

INTERFACE PROGRAM IN BASIC

```
100 REM ASM09- EXEC. FOR ASM09.BIN
200 CLEAR
300 BL=1:REM DEL 10 FOR NO BLOADS
400 HOME:VTAB(4):INVERSE
500 D$=CHR$(4):PRINTD$;"PR#0"
600 PRINTD$;"CLOSE"
700 X$=" THE MILL / APPLE ][ 6809 ASSEMBLER "
800 X=LEN(X$)
900 PRINTSPC(X):PRINT:PRINTX$:PRINTSPC(X):PRINT
1000 X$=" (C) 1981 CONEJO COMPUTER PRODUCTS  "
1100 PRINTSPC(X):PRINT:PRINTX$:PRINTSPC(X):PRINT
1200 REM --------------------
1300 REM ---- CONFIGURATION PARAMS
1400 REM PRINTER PAGE DIM.
1500 WD=78:DP=66:REM WIDTH,DEPTH
1505 REM *** MEMORY & I/O CONSTANTS ***
1510 MEM = 8192: REM>> LOAD HERE <<
1700 IO=MEM:REM 6502 CODE HERE
1710 CODE=IO+256:REM 6809 CODE HERE
1800 SLOT=4:REM MILL SLOT#
1810 SN=SLOT*16:REM LSB OF I/O ADDR
1900 SLOT=49280+SN:REM SLOT->ADDR
2000 REM >>> 4 PAGE ZERO BYTES <<<
2100 RCMD=250:REM MAILBOX W/09 & ASM09IO.BIN
2200 TCMD=RCMD+2:REM DITTO
2210 POKE TCMD+2,SN:REM USED BY ASM09IO
2220 XFADR=512:REM $200, SEE XFILE
2300 REM --- END OF CONSTANTS
2400 REM -------------------
2500 LI=0:OB=0:F1$="":PASS=1
2600 REM CRT DIM.
2800 REM GET '09 CODE
2900 ONERR GOTO 3200
3000 GOSUB16000
3100 GOSUB 15000:GOTO 3600
3200 HOME:FLASH
3300 PRINT"ASM09.BIN OR ASM09IO.BIN NOT FOUND"
3400 GOTO3400
3500 REM ----
3600 HOME
3610 VTAB(4):INVERSE
3620 PRINT" SELECT DRIVE NUMBER, <CR> FOR #1 ";
3630 NORMAL:PRINT" ";CHR$(7);:GET Y$
3640 IF ASC(Y$)=13 THEN Y$="1"
3650 X=ASC(Y$):IF (X<=48)OR(X>50)THEN3600
3655 VTAB(4):HTAB(37):PRINTY$
3660 DR$=",D"+Y$
3665 REM DEFAULT LISTING SIZE:
3670 POKE CODE+3,38:POKE CODE+4,23
3680 REM <PRINTER W/O FORM FEED> POKE CODE+6,0
3700 VTAB(8):PRINT
3800 VTAB(8):INVERSE
3900 PRINT"ENTER SOURCE FILE NAME";CHR$(7)
4000 VTAB(8):HTAB(25)
4100 NORMAL:INPUT F$:GOSUB 15000
4200 PRINT
4300 IF F$="" THEN 3600
4400 O$=F$+".HEX"
4500 ONERR GOTO 4900
4600 F$=F$:PRINTD$;"VERIFY ";F$+DR$
4700 PRINTD$;"READ ";F$:PRINTD$;"CLOSE"
4800 GOTO 5200
4900 VTAB(10):PRINTSPC(39):VTAB(10):HTAB(1)
5000 FLASH:PRINT"TEXT FILE: ";
```

```
5100 PRINTF$+DR$;" NOT FOUND":GOTO3700
5200 GOSUB 15000:VTAB(10):PRINTSPC(39):PRINT
5300 VTAB(13):INVERSE
5400 PRINT"WANT OBJECT CODE (Y/N)";
5500 GOSUB 15300:OB=Y
5600 IFY=-1 THEN 5300
5700 IF OB=0 THEN 6000
5800 ON ERR GOTO 5910
5900 PRINTD$;"DELETE ";O$
5910 GOSUB15000
6000 VTAB(15):INVERSE
6100 PRINT"    WANT LISTING (Y/N)";
6200 GOSUB15300:LI=Y
6300 IFY=-1 THEN 6000
6400 POKE CODE+2,((OB*128)+LI):REM TO ASM09.BIN
6500 IF LI=0 THEN 7700
6600 VTAB(17):INVERSE
6700 PRINT"SELECT LISTING DEVICE:"
6800 PRINT" <CR> = CRT, ";
6900 PRINT"# = SLOT # OF PRINTER";
7000 NORMAL:PRINT" ?";CHR$(7);:GET LI$
7100 PRINT LI$
7200 IF ASC(LI$)=13 THEN LI$="0":GOTO7600
7300 X=ASC(LI$)
7400 IF (X<48)OR(X>55)THEN6600
7500 POKE CODE+3,WD:POKE CODE+4,DP
7600 INVERSE
7700 VTAB(23):PRINT"PASS 1, FILE: ";F$
7800 NORMAL
7900 GOSUB 13100:REM STARTUP MILL
8000 REM !!!PRINTD$;"OPEN ";F$
8100 PRINTD$;"READ ";F$
8110 FX$=F$
8200 REM:--POLL--
8400 GOTO8600
8500 REM *** (PREMATURE) EOF
8505 REM SEND -1 DATA TO ASSEMBLER
8510 GOSUB 15000
8520 POKE RCMD+1,255:POKE RCMD,0
8600 REM ---------------------
8605 ONERR GOTO 8500
8610 CALL IO:REM GOTO 6502 CODE
8700 GOSUB15000
8800 RX=PEEK(RCMD):TX=PEEK(TCMD)
8900 REM RCMD:1=OPEN,2=READ,3=REWIND
8910 REM      4=STOP,5=OPEN NEW,6=RESUME MAIN
9000 REM TCMD:1=LIST,2=BEGIN OBJ,3=BEGIN LIST
9100 IF TX THEN GOSUB 11200
9200 IF RX>6 THEN STOP
9300 ON RX GOSUB 10900,10800,9600,12500,10610,10700
9400 POKE RCMD,0
9500 GOTO 8200
9600 REM *** REWIND (3)
9700 PRINT:PRINTD$;"CLOSE"
9800 IF LI$="0" THEN HOME:INVERSE
9900 PRINT"PASS 2, OBJECT FILE=";:NORMAL
10000 IF OB=0 THENPRINT"(NONE)":GOTO 10300
10100 PRINT" ";O$
10200 GOSUB15000:PRINTD$;"OPEN ";O$
10300 PRINTD$;"PR# ";LI$
10310 FX$=F$
10400 PRINTD$;"READ ";FX$
10500 POKE TCMD,0:PASS=2
10600 RETURN
```

```
10610 REM *** READ AN XFILE (5)
10620 X=XFADR:F1$="":POKE RCMD+1,0
10630 X1=PEEK(X):IF X1=0 THEN 10650
10640 F1$=F1$+CHR$(X1):X=X+1:GOTO 10630
10650 ON ERR GOTO 10690
10652 IF F1$<>""THENPRINTD$;"VERIFY ";F1$:FX$=F1$
10655 GOSUB15000
10666 PRINTD$;"READ ";FX$
10670 RETURN
10690 GOSUB 15000:POKE RCMD+1,255:REM ERROR
10692 PRINT
10693 PRINTF1$;" NOT FOUND";CHR$(7)
10696 FX$=F$:F1$="":RESUME
10700 REM *** RESUME READ OF MAIN (6)
10710 PRINTD$;"CLOSE ";FX$
10720 FX$=F$:F1$=""
10730 PRINTD$;"READ ";FX$
10740 RETURN
10800 REM *** READ (2)
10810 STOP
10900 REM *** OPEN (1)
11000 RETURN
11100 REM -------
11200 REM TCMD ACTION
11300 ON TX GOTO 11500,11600,11900
11400 STOP:REM TX>3
11500 STOP:REM TX=1 DONE IN ASM09IO
11600 REM TX=2 START HEX
11700 PRINTD$:REM GOOD OLE DOS!
11800 PRINTD$;"WRITE ";O$:GOTO 12200
11900 REM TX=3=END HEX
12000 PRINT
12100 PRINTD$;"READ ";FX$
12200 POKE TCMD,0
12300 RETURN
12400 REM ------
12500 REM END PASS 2
12600 PRINTD$;"PR# 0"
12700 PRINTD$;"CLOSE"
12800 POKE SLOT+2,0
12900 END
13000 REM
13100 REM INITIALIZE THE MILL
13200 POKE RCMD,0:POKE TCMD,0
13300 B8=128
13400 POKE SLOT+0,B8
13500 POKE SLOT+1,B8
13600 POKE SLOT+2,0
13700 POKE SLOT+3,B8
13800 POKE SLOT+4,B8
13900 POKE SLOT+5,B8
14000 POKE SLOT+7,B8
14100 REM SETUP 09'S RESTART
14200 REM  JMP CODE
14300 REM  AT $40FA
14400 X=PEEK(65534)*256+PEEK(65535)
14500 POKE X+0,126:REM $7E
14600 POKE X+1,CODE/256
14700 POKE X+2,CODE-(256*(CODE/256))
14800 REM ASM09IO STARTS UP '09
14900 RETURN
15000 REM *** KILL ON ERR
15100 POKE 216,0
15200 RETURN
```

```
15300 REM GET Y/N REPLY
15400 Y=-1
15500 NORMAL:PRINT" ?";CHR$(7);:GET Y$
15600 PRINTY$;
15700 IF Y$="Y" THEN Y=1
15800 IF Y$="N" THEN Y=0
15900 PRINT:RETURN
16000 REM STARTUP
16700 IF BL=0 THEN FOR I=1TO500:NEXT I:RETURN
16900 VTAB(22):HTAB(8)
17000 FLASH:PRINT"STANDBY- BLOAD RUNNING"
17100 NORMAL
17200 PRINTD$;"BLOAD ASM09.BIN,A",STR$(CODE)
17300 PRINTD$;"BLOAD ASM09IO.BIN,A"STR$(IO)
17400 RETURN
17500 REM LAST LINE
```

ASM09IO

INTERFACE SUBROUTINE FOR ASM09

```
SOURCE FILE: ASM09IO
0000:               *
0000:               ************************************
0000:               * (C) 1981, CONEJO COMPUTER PRODUCTS
0000:               * 3655 THOUSAND OAKS BL.
0000:               * WESTLAKE VILLAGE, CA   91362
0000:               *   ALL RIGHTS RESERVED
0000:               *
0000:               ******* ASM09IO.TXT, 6502 CODE ******
0000:               *
0000:               *  >> THIS 6502 CODE IS POSITION-INDEPENDENT
0000:               *     AND CAN BE RELOCATED AT "BLOAD"-TIME
0000:               *
0000:               * THIS PROGRAM INTERFACES ASM09, THE BASIC
0000:               * CODE TO ASM09.BIN, THE 6809 CODE
0000:               *
2000:               CODEAT   EQU  8192       WHERE ASM09.BIN GOES
----- NEXT OBJECT FILE NAME IS ASM09IO.OBJ0
1E00:                        ORG  CODEAT-512 WHERE ASM09IO.BIN GOES
1E00:               *
1E00:               * DOS ADDRESSES:
1E00:               *
FDED:               COUT     EQU  $FDED      CHAR OUT
FD0C:               CIN      EQU  $FD0C      CHAR IN
1E00:               *
1E00:               * PAGE ZERO MAILBOXES W/BASIC AND ASM09
1E00:               *
00FA:               RXCMD    EQU  250        MAILBOX W/6809
00FC:               TXCMD    EQU  RXCMD+2    DITTO
1E00:               * ASM09 POKES SLOT# * 16 HERE:
00FE:               SLOTNO   EQU  TXCMD+2    BASIC POKES IT HERE
1E00:               * EG: IF MILL IN SLOT 4, C(SLOTNO)=$40
1E00:               *
1E00:               * REGISTERS WITHIN THE MILL
1E00:               *
C080:               SLOT0    EQU  $C080      ADDR OF SLOT 0
C081:               RUN09    EQU  SLOT0+1    IO ADDR+2 IS RUN BIT
C082:               RESET    EQU  SLOT0+2    6809 RESET, 1=FALSE
1E00:               *
1E00:               * PROTOCOL CODES USED HEREIN, OTHERS IN BASIC09
1E00:               *
0001:               LISTCH   EQU  1          SEND CHAR TO LISTING
0002:               READCH   EQU  2          READ CHAR FROM DISK
1E00:               *
1E00:               ***********************************
1E00:               * BASIC DOES A "CALL IO" TO HERE: *
1E00:               ***********************************
1E00:               *
1E00:A4 FE                   LDY  SLOTNO     GET IOADDRESS LOW PART
1E02:A9 80                   LDA  #$80
1E04:99 82 C0                STA  RESET,Y    SET RESET=FALSE
1E07:99 81 C0                STA  RUN09,Y    SET RUN=1
1E0A:               *
1E0A:A2 00          A000     LDX  #0         CONSTANT
1E0C:A4 FE                   LDY  SLOTNO     GET SLOT NO.
1E0E:A9 80                   LDA  #$80       SET RUN=1
1E10:99 81 C0                STA  RUN09,Y
1E13:               *
1E13:               *
1E13:               *
1E13:               *
```

```
1E13:           *
1E13:           *
1E13:           *
1E13:           *
1E13:           *
1E13:           *
1E13:A5 FA      A100    LDA  RXCMD      ANYTHING ON RCV?
1E15:F0 17              BEQ  B100       IF NO
1E17:C9 02              CMP  #READCH    IS IT "READ A CHARACTER" ?
1E19:F0 05              BEQ  A200       IF SO, DO IT
1E1B:           * ALL OTHER COMMANDS HANDLED BY BASIC
1E1B:8A                 TXA             ZERO
1E1C:99 81 C0           STA  RUN09,Y    HALT THE 09
1E1F:60                 RTS             RETURN TO BASIC
1E20:           *
1E20:           * IS READ A CHAR FROM SOURCE FILE CMD
1E20:           *
1E20:8A         A200    TXA             ZERO
1E21:99 81 C0           STA  RUN09,Y    HALT THE 09
1E24:20 0C FD           JSR  CIN        FETCH CHAR
1E27:85 FB              STA  RXCMD+1    PASS VIA MAILBOX
1E29:86 FA              STX  RXCMD      RESET COMMAND
1E2B:18                 CLC
1E2C:90 DC              BCC  A000       (JMP) NEXT COMMAND
1E2E:           *
1E2E:           * POLL THE OTHER SIDE, OUTPUT FROM THE 09
1E2E:           *
1E2E:A5 FC      B100    LDA  TXCMD      ANY COMMAND?
1E30:F0 E1              BEQ  A100       IF NO
1E32:C9 01              CMP  #L1STCH     SEND TO LISTING?
1E34:D0 10              BNE  B300       IF NO
1E36:           *
1E36:           * SEND LISTING CHARACTER
1E36:           *
1E36:8A                 TXA             ZERO
1E37:99 81 C0           STA  RUN09,Y    STOP THE 09
1E3A:           *   IN CASE I/O NEEDS FULL SPEED 6502
1E3A:A5 FD              LDA  TXCMD+1    GET DATA
1E3C:09 80              ORA  #$80       MERGE MSB
1E3E:20 ED FD           JSR  COUT       SEND CHAR
1E41:86 FC              STX  TXCMD      RESET COMMAND
1E43:18                 CLC
1E44:90 C4              BCC  A000       CONTINUE POLL
1E46:           *
1E46:           * IS START/STOP OBJECT
1E46:           *
1E46:8A         B300    TXA
1E47:99 81 C0           STA  RUN09,Y    STOP '09
1E4A:60                 RTS             GOTO BASIC
1E4B:           *
1E4B:           *
1E4B:           ***** END OF TEXT ******

*** SUCCESSFUL ASSEMBLY: NO ERRORS
```

LOAD09

6502 PROGRAM TO LOAD PROGRAMS FOR THE MILL

```
SOURCE FILE: LOAD09
0000:             * LOAD09 - 6502 M/L
0000:             * FILENAME=LOAD09.BIN
0000:             * REV: 5/9/81
0000:             *
----- NEXT OBJECT FILE NAME IS LOAD09.OBJ0
6000:                      ORG   $6000      <<< CHANGE AS NEEDED <<<
6000:             *
6000:             * THE CALLER OF THIS SUBROUTINE MUST
6000:             * OPEN AN ASM09 HEX FILE AND ISSUE A READ
6000:             * COMMAND TO DOS. THIS CODE READS ALL DATA
6000:             * RECORDS, STORES ALL DATA, AND RETURNS
6000:             * AFTER PROCESSING THE "END" RECORD.
6000:             * ON RETURN, THE END RECORD'S STARTING
6000:             * ADDRESS IS IN LOCATIONS .
00FA:             XFERHI   EQU   250        MOST  SIGNIF. BITS
00FB:             XFERLO   EQU   251        LEAST SIGNIF. BITS
6000:             *
6000:             *
FDOC:             INCH     EQU   $FDOC      READ CHAR FROM INPUT
6000:             *
6000:D8           START    CLD              NO DECIMAL
6001:             *
6001:20 0C FD     A100     JSR   INCH       GET CHAR
6004:29 7F                 AND   #$7F
6006:C9 3A                 CMP   #$3A       ":", START OF RECORD
6008:D0 F7                 BNE   A100
600A:20 48 60              JSR   HEX2       READ RECORD TYPE
600D:C9 03                 CMP   #3         ="END"
600F:F0 2C                 BEQ   B200       IF END RECORD
6011:C9 01                 CMP   #1         ="DATA"
6013:F0 04                 BEQ   B100
6015:00                    BRK              ILLEGAL RECORD
6016:4C 00 60              JMP   START
6019:             *
6019:             * PROCESS DATA RECORD
6019:             *
6019:20 48 60     B100     JSR   HEX2       GET ADDRESS-HI
601C:8D 2E 60              STA   B160+2     SETUP INDIRECT
601F:20 48 60              JSR   HEX2       GET ADDRESS-LO
6022:8D 2D 60              STA   B160+1
6025:20 48 60              JSR   HEX2       GET COUNT
6028:AA                    TAX              SAVE COUNT
6029:20 48 60     B150     JSR   HEX2       GET DATA BYTE
602C:8D FF FF     B160     STA   $FFFF      STORE DATAA
602F:EE 2D 60              INC   B160+1     ADVANCE POINTER
6032:D0 03                 BNE   B170
6034:EE 2E 60              INC   B160+2     16 BIT ADDR
6037:CA           B170     DEX              LOOP ON RECORD COUNT
6038:D0 EF                 BNE   B150
603A:4C 01 60              JMP   A100       NEXT RECORD
603D:             *
603D:             * PROCESS END RECORD
603D:             *
603D:20 48 60     B200     JSR   HEX2       GET HI OF STARTING ADDRESS
6040:85 FA                 STA   XFERHI     STORE IT
6042:20 48 60              JSR   HEX2       GET LOW OF STARTING ADDRESS
6045:85 FB                 STA   XFERLO     STORE IT
6047:             *
6047:             *
6047:             *
```

```
6047:               *
6047:               *
6047:               *
6047:               *
6047:               *
6047:               *
6047:               *
6047:               * RETURN TO CALLER
6047:               *
6047:60                      RTS
6048:               *
6048:               ***********************************
6048:               *
6048:               * READ TWO HEX/ASCII BYTES, RETURN
6048:               * BINARY EQUIVALENT IN Y.
6048:               *
6048:20 5A 60       HEX2     JSR  HEX1        READ 1ST
604B:0A                      ASL  A           MSB
604C:0A                      ASL  A
604D:0A                      ASL  A
604E:0A                      ASL  A
604F:8D 59 60                STA  HEX2X
6052:20 5A 60                JSR  HEX1        GET 2ND
6055:0D 59 60                ORA  HEX2X       MERGE
6058:60                      RTS
6059:00             HEX2X    DFB  0
605A:               *
605A:               * READ AND CONVERT 1 HEX CHAR
605A:               * RETURN BINARY EQUIVALENT IN ACCUM.
605A:               *
605A:20 0C FD       HEX1     JSR  INCH        GET CHAR
605D:29 7F                   AND  #$7F        KILL MSB
605F:C9 3A                   CMP  #$39+1      '9'+1
6061:30 03                   BMI  HEX1A       IF 0..9
6063:38                      SEC
6064:E9 37                   SBC  #55         MAKE 10..15
6066:29 0F          HEX1A    AND  #$0F        4 BITS
6068:60                      RTS              IN ACCUM
6069:               *
6069:               * LAST LINE
6069:               *

*** SUCCESSFUL ASSEMBLY: NO ERRORS
```

```
6001 A100          6019 B100          6029 B150          602C B160
6037 B170          603D B200          605A HEX1          6066 HEX1A
6048 HEX2          6059 HEX2X         FD0C INCH          6000 START
  FA XFERHI          FB XFERLO
```

SAMPLE OF LOADER FORMAT

```
011000022002

:0110023A0628328D016AAE8CF7318D0065A6A0A78026FAC630318D00676C26A62681392341E7266
C25A62581392337E7256C24A6248139232DE7247DC030

:01103C386C23A62381392320E7236C22A62281392316E7226C21A6218139230CE7216CA4A6A4813
92302E7A4308D000C10AE8C96A680A7A026FA20A1

:0110740E3638303920434F554E54494E4720

:0110820830303030303000

:031000

]
```

DEMO1

LISTING

DEMO1.TXT - A SIMPLE DEMONSTRATION OF 6809+APPLE
(C) 1981, CONEJO COMPUTER PRODUCTS

```
                  * FILE=DEMO1.TXT
                  * 6809 / APPLE DEMO PROG
                  * RUNS A COUNTER ON TOP LINE
                                OPT    NUM
1000                            ORG    4096        <<< CHANGE AS DESIRED <<<
          07F0     SCREEN       EQU    $7F0        CRT ADDRESS
          C030     BEEP         EQU    $C030       I/O FOR SPEAKER
1000 2002          START        BRA    BEGIN
1002 0628          ADDR         FDB    $628        SCREEN ADDR

1004 328D016A      BEGIN        LEAS   END+232,PCR SET STACK POINTER
1008 AE8CF7                     LDX    ADDR,PCR    SCREEN ADDRESS
100B 318D0065                   LEAY   MSG1,PCR    "6809 COUNTING ...."
100F A6A0          L100         LDA    ,Y+
1011 A780                       STA    ,X+
1013 26FA                       BNE    L100

1015 C630          L200         LDB    #'0         ASCII ZERO
1017 318D0067                   LEAY   MSG1A,PCR   BUFFER ADDRESS
101B 6C26                       INC    6,Y         FROM HERE WE INCREMENT BCD DIGITS
101D A626                       LDA    6,Y         BY ADDING 1 TIL '9' THEN DO CARRY
101F 8139                       CMPA   #'9
1021 2341                       BLS    L300
1023 E726                       STB    6,Y
1025 6C25                       INC    5,Y
1027 A625                       LDA    5,Y
1029 8139                       CMPA   #'9
102B 2337                       BLS    L300
102D E725                       STB    5,Y
102F 6C24                       INC    4,Y
1031 A624                       LDA    4,Y
1033 8139                       CMPA   #'9
1035 232D                       BLS    L300
1037 E724                       STB    4,Y
1039 7DC030                     TST    BEEP        CLICK SPEAKER
103C 6C23                       INC    3,Y
103E A623                       LDA    3,Y
1040 8139                       CMPA   #'9
1042 2320                       BLS    L300
1044 E723                       STB    3,Y
1046 6C22                       INC    2,Y
1048 A622                       LDA    2,Y
104A 8139                       CMPA   #'9
104C 2316                       BLS    L300
104E E722                       STB    2,Y
1050 6C21                       INC    1,Y
1052 A621                       LDA    1,Y
1054 8139                       CMPA   #'9
1056 230C                       BLS    L300
1058 E721                       STB    1,Y
105A 6CA4                       INC    ,Y
105C A6A4                       LDA    ,Y
105E 8139                       CMPA   #'9
1060 2302                       BLS    L300
1062 E7A4                       STB    ,Y

1064 308D000C      L300         LEAX   MSG1,PCR    SEND DATA TO CRT
1068 10AE8C96                   LDY    ADDR,PCR
```

```
106C A680      L400  LDA    ,X+        MOVE LOCAL TO CRT MEMORY
106E A7A0            STA    ,Y+
1070 26FA            BNE    L400
1072 20A1            BRA    L200       MAIN LOOP

1074 3638303920 MSG1 FCC    "6809 COUNTING "
107A 434F554E54494E47
1082 20
1082 3030303030 MSG1A FCC   "0000000",0
1088 303000
          000E  CNT  EQU    MSG1A-MSG1
          108A  END  EQU    *

          1000       END    START
```

```
DEMO1.TXT - A SIMPLE DEMONSTRATION OF 6809+APPLE
(C) 1981, CONEJO COMPUTER PRODUCTS

   13 SYMBOLS IN TABLE:

ADDR   $1002  BEEP   =C030  BEGIN $1004  CNT    =000E  END    =108A  L100   $100F
L200   $1015  L300   $1064  L400   $106C  MSG1   $1074  MSG1A $1082  SCREEN=07F0
START  $1000

SYMBOL TABLE END: 417D

    0 STATEMENT ERROR(S), LAST PC:1089
```

STELLATION TWO

NOTICE

Stellation Two and Conejo Computer Products reserve the right to make improvements in the Assembler Development Kit at any time and without notice.

DISCLAIMER OF ALL LIABILITY AND WARRANTIES:

COPYRIGHT:

-- OWNER'S REGISTRATION FORM --

CONEJO COMPUTER PRODUCTS
3655 Thousand Oaks Blvd. Suite 255
Westlake Village, California 91362 USA

THE MILL, 6809 ASSEMBLER PKG., V1

NAME --

FIRM --

STREET --

CITY/STATE --ZIP--------

COUNTRY --

DATE PURCHASED: MONTH:---------- DAY:----, YEAR: 19--

DEALER --

THE MILL SERIAL NUMBER --

APPLICATION, please check one:

OEM [] INDUSTRIAL END-USER [] SCHOOL [] HOBBY []

SOFTWARE DEVELOPER [] OTHER [] -------------------------

We would appreciate a brief explanation of your intended application and any comments regarding the manual, sugestions for new products, etc.: